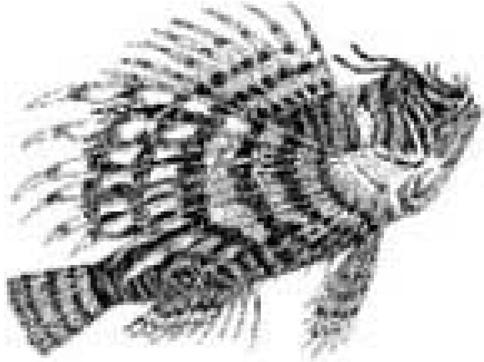
Building Web Services in Java

Java™ *and* SOAP



O'REILLY®

Robert Englander

Java and SOAP

Robert Englander

Publisher: O'Reilly

Edition May 2002

ISBN: 0-596-00175-4, 276 pages

Java™ and SOAP provides Java developers with an in-depth look at SOAP (the Simple Object Access Protocol). Of course, it covers the basics: what SOAP is, why it's soared to a spot on the Buzzwords' Top Ten list, and what its features and capabilities are. And it shows you how to work with some of the more common Java APIs in the SOAP world: Apache SOAP and GLUE.

Java™ and SOAP also discusses interoperability between the major SOAP platforms, including Microsoft's .NET, SOAP messaging, SOAP attachments, message routing, and a preview of the forthcoming AXIS APIs and server. If you're a Java developer who would like to start working with SOAP, this is the book you need to get going.

Dedication

Once again, for my daughter Jessica.

Preface

The Simple Object Access Protocol, or SOAP, is the latest in a long line of technologies for distributed computing. It differs from other distributed computing technologies in that it is based on XML, and also that thus far it has not attempted to redefine the computing world. Instead, the SOAP specification describes important aspects of data content and structure as they relate to familiar programming models like remote procedure calls (RPCs) and message passing systems.

These specifications live squarely in the world of XML. SOAP is not bound to a specific programming language, computing platform, or software development environment. There are SOAP implementations that provide bindings for a variety of programming languages like C#, Perl, and Java™. Without these implementations SOAP remains in the abstract: a great concept without manifestation. It is the binding to software development languages that makes SOAP come alive, and that is what this book is about. Java is a natural for XML processing, making it perfect for building SOAP services and client applications. If building SOAP-aware software in Java is what you want to do, this book is just what you need to get started.

Intended Audience

This book is for everyone interested in how to access SOAP-based web services in Java, as well as how to build SOAP-based services in Java. It's written for programmers, students, and professionals who are already familiar with Java, so it doesn't spend any time covering the basic concepts or syntax of the language. If you aren't familiar with Java, you may want to keep a copy of a Java language book, like O'Reilly's *Learning Java* or *Java in a Nutshell,* close by.

A Moment in Time

The SOAP specification is still evolving. This book describes SOAP according to Version 1.1 of the spec. Although the concepts and techniques covered should continue to be relevant in future SOAP releases, there will certainly be important additions to SOAP as new versions of the spec are finalized. The Java implementations we'll be looking at will continue to evolve as well. Obviously, the descriptions and examples in this book will become dated or even obsolete over time — and that time will probably be sooner rather than later, given the speed at which web services are evolving. In fact, the handwriting is already on the wall: Apache SOAP Version 2, on which many of the examples are based, is destined to be replaced by Apache SOAP 3 (also known as Axis), which is currently available in an early release and is discussed briefly in Chapter 9. Axis, in turn, is committed to supporting the JAX RPC and JAXM API specifications, which are themselves still under development. An early access release of the reference implementation for these specifications is available from Sun Microsystems (and discussed in Chapter 11); this release is more recent than the most recent release of Axis. And it would be foolish to think that the JAX Pack specifications will mark the end of the evolutionary process. However, when the inevitable happens, you'll be armed with the knowledge and understanding necessary to keep pace with the changes.

How This Book Is Organized

The chapters in this book are organized so that each one builds upon the information presented in previous chapters, so it's best if you read the chapters in order.

Chapter 1

This chapter provides an overview of SOAP, including related technologies, problem spaces, and comparisons to other solutions. It also introduces Apache SOAP and GLUE, the SOAP implementations that will be used throughout the book.

Chapter 2

This chapter describes the SOAP Envelope, a structured XML document that carries the payload of a SOAP transaction between client and server. It covers all aspects of a SOAP Envelope, including Headers, SOAP Body elements, and Faults. Some details of the SOAP HTTP binding are also included.

Chapter 3

This chapter covers the data encoding of a SOAP transaction, including rules for encoding and serializing data elements. It starts out with a description of namespaces, and then delves into the serialization of both simple and complex data types.

Chapter 4

This chapter goes deep into SOAP-based remote procedure call (RPC) style services. Extensive coverage of service methods and parameters is provided, along with the details of service deployment and activation mechanisms.

Chapter 5

This chapter looks at the creation of services with complex method parameters and return values such as arrays and Java beans. It covers the mechanisms available for mapping these types to Java classes on both client and server systems.

Chapter 6

This chapter covers the use of nonstandard custom data types, picking up where Chapter 5 left off. It looks at some of the tools and APIs used to pass instances of custom data types as parameters and return values. It also details the techniques of writing Java classes for serializing and deserializing custom types.

Chapter 7

This chapter describes SOAP Faults, along with their relationship to Java exceptions. It looks at the default mechanisms provided, as well as techniques for generating and extending the contents of Faults.

Chapter 8

This chapter starts out by describing the use of SOAP message-style services, an alternative to the RPC model. It also looks at passing literal XML inside of a SOAP Envelope, and finishes up with a look at SOAP Attachments.

Chapter 9

This chapter looks at getting SOAP clients and servers, developed using different technologies, to work properly together. An introduction to the Web Services Description Language (WSDL) is provided. Examples are developed that cover clients and services built using Apache SOAP and GLUE, a sneak peek at Apache Axis, and Java clients accessing Microsoft .NET services.

Chapter 10

This chapter looks at the use of SOAP Headers, which provide a means to pass data between clients and services that lie outside the scope of the SOAP Body. It covers the development of an intermediary service that acts as a message router to another service. Some Java classes are developed for extending the Apache SOAP framework in order to work with SOAP Headers.

Chapter 11

This chapter examines the emerging standard: the Java API for XML-based RPC (JAX-RPC). It's a look at an early release of Sun's reference implementation. This chapter covers the development of both a service and a client, and also looks at using the tools to develop code for accessing services described by WSDL. A final commentary on JAXM is also included.

Conventions Used in This Book

`Constant Width` is used for:

- Anything that might appear in a Java program, including keywords, operators, data types, constants, method names, variable names, class names, interface names, and Java package names.
- Command lines and options that should be typed verbatim on the screen.
- Namespaces.

Italic is used for:

- Pathnames, filenames, and Internet addresses, such as domain names and URLs. Italics is also used for executable files.

Making fine distinctions in a book like this is generally a losing battle. But I have tried to distinguish between namespaces (constant width) and URLs (italic), even though they look identical. Likewise, I've tried to distinguish between Java methods (constant width and ending in a pair of parentheses) and the methods exported by the SOAP service (constant width, no parentheses).

This icon signifies a note relating to the nearby text.

This icon signifies a warning relating to the nearby text.

How to Contact Us

I've certainly tried to be accurate in my descriptions and examples, but errors and omissions will inevitably exist. If you find mistakes, or you think I've left out important details, or you'd like to contact me for some other reason related to this work, you can contact me directly at:

rob@mindstrm.com

Alternately, address comments and questions concerning this book to the publisher:

O'Reilly & Associates, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
(800) 998-9938 (in the United States or Canada)
(707) 829-0515 (international/local)
(707) 829-0104 (fax)

There is a web page for this book, which lists errata, examples, or any additional information. You can access this page at:

http://www.oreilly.com/catalog/javasoap

To comment or ask technical questions about this book, send email to:

bookquestions@oreilly.com

For more information about books, conferences, Resource Centers, and the O'Reilly Network, see the O'Reilly web site at:

http://www.oreilly.com/

Retrieving Examples Online

The code for the examples throughout this book is available online at:

http://www.mindstrm.com/javasoap

Acknowledgments

My good friend Rinaldo DiGiorgio continues, to this day, to keep me interested in Java and its related technologies. I don't think anyone has been a greater influence on my Java work than he has. Thanks, Rinaldo, for keeping me on the right path.

Many thanks go to David Askey and Anne Thomas Manes for reviewing the book and providing valuable feedback. They managed to find errors and offer advice that makes this a better book than it would have been without their help. Thanks to Lorraine Pecorelli for reading every chapter and making sure the words made sense. My deepest appreciation goes to Mike Loukides, the editor of this book. There were many obstacles to getting this project finished, and Mike's commitment and loyalty was key to turning the effort into a book. A thank you also is due to the O'Reilly design and production crew.

And finally, thanks to my family, Jessica and Carolyn, for their support. I'm not going to thank my friends this time — they were no help at all!